COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MÁSTER

DOCUMENT INTELLIGENCE: UNIVERSAL DOCUMENT ANALYSIS

Autor: Pablo García Bolívar

Director: José Luis Gahete Díaz

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

DOCUMENT INTELLIGENCE: UNIVERSAL DOCUMENT ANALYSIS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

BOLIVAR

Fdo.: Pablo García Bolívar Fecha: 13/07/2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: José Luis Gahete Díaz Fecha: 14/07/2025

MÁSTER UNIVERSITARIO EN BIG DATA

TRABAJO FIN DE MÁSTER

DOCUMENT INTELLIGENCE: UNIVERSAL DOCUMENT ANALYSIS

Autor: Pablo García Bolívar

Director: José Luis Gahete Díaz

Madrid

DOCUMENT INTELLIGENCE: ANÁLISIS UNIVERSAL DE DOCUMENTOS

Autor: García Bolívar, Pablo.
Director: Gahete Díaz, José Luis.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Esta tesis presenta una plataforma modular basada en esquemas para la inteligencia documental, capaz de procesar cualquier tipo de documento sin depender de plantillas ni modelos predefinidos. Combina motores OCR, modelos de lenguaje (LLMs) y flujos de trabajo ajustados para permitir una clasificación precisa y una extracción estructurada de datos a nivel de campo, adaptándose a múltiples formatos y casos de uso. Gracias a su infraestructura nativa en la nube, el sistema permite un procesamiento documental escalable, rentable y explicable en entornos empresariales reales. Los casos de uso incluyen ámbitos legales, gestión de multas de tráfico y plataformas internas, validando su precisión, adaptabilidad y preparación para entornos de producción.

Palabras clave: Inteligencia Documental, IA empresarial, LLMs

1. Introducción

La era digital ha acelerado la proliferación de documentos, revelando a su vez un importante cuello de botella: la mayoría de los documentos empresariales siguen sin estar estructurados y poco aprovechados. Esta tesis introduce una plataforma de inteligencia artificial de propósito general que permite la clasificación automática y la extracción estructurada de información mediante esquemas definidos por el usuario y modelos de lenguaje. A diferencia de las soluciones basadas en reglas o plantillas fijas, el sistema propuesto admite despliegues dinámicos, escalables y agnósticos al dominio.

2. Análisis de Mercado

El estudio de mercado confirma una demanda creciente en sectores como finanzas, legal y logística por soluciones de procesamiento documental flexibles y asequibles. Un análisis comparativo de los principales competidores (Extend, TableFlow, Invofox, Reducto, Nanonets) revela carencias importantes en cuanto a precios, portabilidad de esquemas y soporte de ajuste fino. Estos vacíos respaldan la dirección técnica de la tesis: un sistema configurable mediante esquemas, desplegable en la nube y compatible con múltiples LLMs (M-Files, 2019) (Extend, n.d.) (TableFlow, n.d.) (Invofox, n.d.) (Reducto, n.d.) (Nanonets, n.d.).

3. Requisitos y Principios de Diseño

Tras el análisis de mercado, la arquitectura de la plataforma se ha definido en torno a cuatro áreas clave: requisitos funcionales, técnicos, económicos y sociales. A nivel funcional, admite flujos de trabajo de clasificación y extracción basados en esquemas, ajuste fino de modelos y compatibilidad con múltiples formatos de entrada. Desde la perspectiva técnica, garantiza modularidad, escalabilidad en la nube e integración fluida de LLMs. Económicamente, está optimizado para mantener un coste inferior a un céntimo por documento procesado. Finalmente, el sistema está alineado con los Objetivos de Desarrollo Sostenible de Naciones Unidas, promoviendo la equidad digital y la transparencia en los procesos, especialmente en organizaciones con recursos limitados.

4. Descripción Técnica

La plataforma se construye como un sistema modular compuesto por módulos de OCR, extracción, ajuste fino e infraestructura. Los documentos se digitalizan utilizando AWS Textract o Mistral OCR, en función de la complejidad del diseño (Mistral, n.d.). La extracción se realiza en base a esquemas que definen la estructura, el tipo de contenido y las reglas de validación de los datos a obtener. LangChain se encarga de orquestar las interacciones con los modelos de lenguaje, y la API de ajuste fino de OpenAI permite mejoras específicas para cada dominio. La infraestructura, desplegada en AWS, utiliza pipelines basados en eventos para garantizar rendimiento y escalabilidad, incorporando servicios como Lambda, S3, SQS y DynamoDB para lograr automatización completa y trazabilidad (OpenAI, n.d.) (AWS, n.d.).

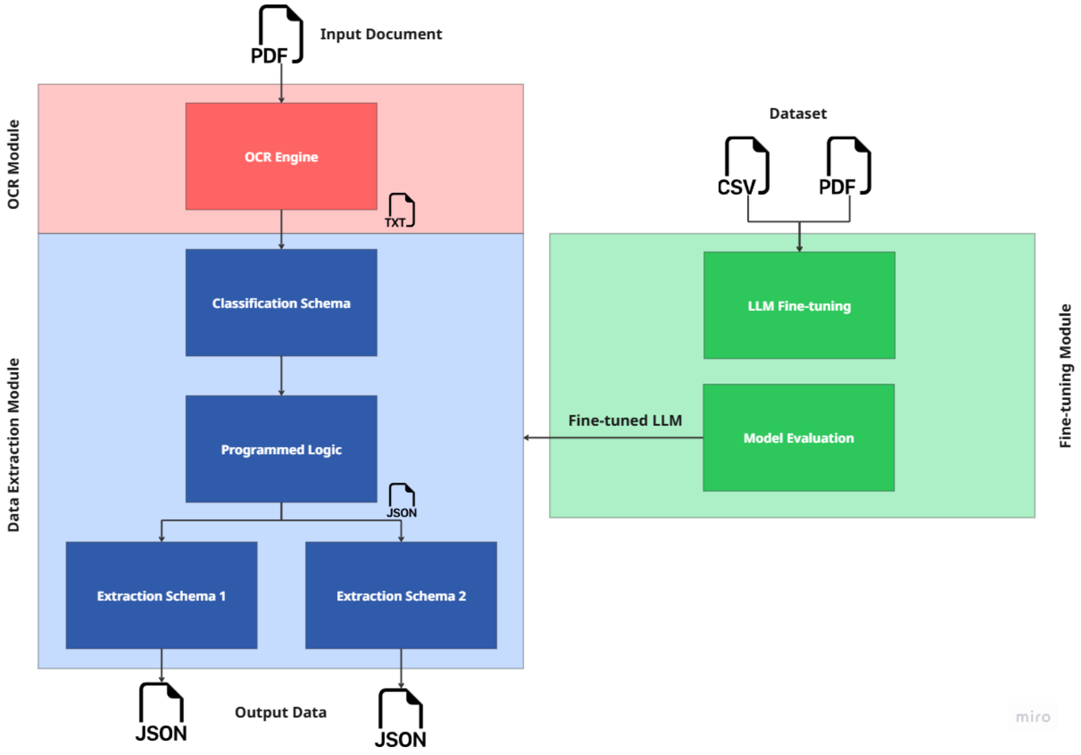


Figura 1.1 Módulos del sistema de Document Intelligence

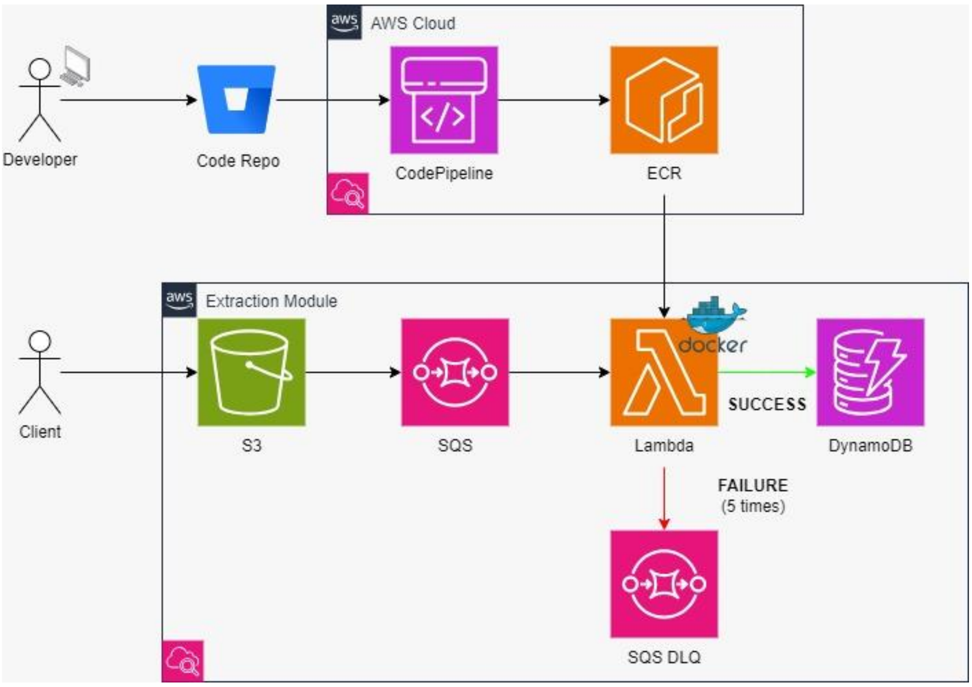


Figura 1.2 Arquitectura de procesamiento documental sobre AWS

5. Integraciones Prácticas en Entornos Empresariales

El sistema se ha implantado en tres contextos de producción distintos para validar su versatilidad y robustez. En una empresa del sector legal especializada en ejecuciones hipotecarias, la herramienta procesó más de 300 tipos documentales, alcanzando una precisión del 87 % en clasificación y del 90 % en extracción de campos clave. Una empresa multinacional de gestión de multas aplicó la plataforma en cinco países y múltiples idiomas, logrando una precisión del 91,3 % en los datos extraídos y reduciendo a la mitad los tiempos de procesamiento. Por último, en una plataforma interna (WorkOps), los usuarios finales pudieron definir e implementar esquemas personalizados mediante una interfaz gráfica, lo que demuestra la accesibilidad y adaptabilidad del sistema incluso en entornos low-code.

6. Conclusiones y Trabajo Futuro

Si bien construir un prototipo funcional de un sistema de inteligencia documental es factible con herramientas modernas, desarrollar una solución lista para producción requiere un enfoque riguroso en robustez, seguridad y usabilidad. Este proyecto demuestra que aspectos como la validación humana, la trazabilidad y la configuración basada en esquemas no son concesiones, sino principios esenciales del diseño en IA empresarial. La arquitectura modular, el enfoque en costes y la posibilidad de ajuste fino convierten a esta plataforma en una propuesta viable para la automatización documental escalable y explicable en múltiples sectores.

7. Referencias

AWS. (s.f.). AWS Products. Obtenido de https://aws.amazon.com/es/

Extend. (s.f.). Extend AI. Obtenido de https://www.extend.ai/

Invofox. (s.f.). Invofox. Obtenido de https://www.invofox.com/es

M-Files. (2019). M-Files. Obtenido de https://www.m-files.com/wp-content/uploads/2023/07/ebook-2019-intelligent-information-management-benchmark-en.pdf.pdf

Mistral. (s.f.). Mistral. Obtenido de https://mistral.ai/news/mistral-ocr

Nanonets. (s.f.). Nanonets. Obtenido de https://nanonets.com/

OpenAI. (s.f.). OpenAI API Docs - GPT Models. Recuperado el 27 de 06 de 2023, de https://platform.openai.com/docs/guides/gpt

Pydantic. (s.f.). Pydantic. Obtenido de https://docs.pydantic.dev/latest/

Reducto. (s.f.). Reducto AI. Obtenido de https://reducto.ai/

TableFlow. (s.f.). TableFlow. Obtenido de https://tableflow.com/

DOCUMENT INTELLIGENCE: UNIVERSAL DOCUMENT ANALYSIS

Author: García Bolívar, Pablo.
Supervisor: Gahete Díaz, José Luis.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This thesis presents a modular, schema-based platform for Document Intelligence—capable of processing any document type without relying on templates or predefined models. It combines OCR engines, LLMs, and fine-tuned pipelines to enable accurate classification and field-level data extraction across diverse formats and use cases. Backed by cloud-native infrastructure, the system achieves scalable, cost-effective, and explainable document processing in real-world enterprise deployments. Use cases include legal, traffic, and low-code environments, validating the tool's accuracy, adaptability, and production readiness.

Keywords (3): Document Intelligence, Enterprise AI, LLMs

1. Introduction

The digital era has accelerated document proliferation while exposing a critical bottleneck: most business documents remain unstructured and underutilized. This work introduces a general-purpose AI platform that enables automated classification and structured data extraction from documents, leveraging user-defined schemas and LLMs. Unlike rule-based or template-bound solutions, the proposed system supports dynamic, scalable, and domain-agnostic deployments.

2. Market Analysis

Market research confirms growing demand across industries—finance, legal, logistics—for flexible and affordable document processing. A comparative analysis of leading competitors (Extend, TableFlow, Invofox, Reducto, Nanonets) reveals gaps in pricing, schema portability, and fine-tuning support. This validates the thesis' technical direction: a system that is schema-configurable, cloud-native, and LLM-agnostic (M-Files, 2019) (Extend, n.d.) (TableFlow, n.d.) (Invofox, n.d.) (Reducto, n.d.) (Nanonets, n.d.).

3. Requirements and Design Principles

Following the market analysis, the platform's architecture was shaped around four core requirement areas: functional, technical, economic, and social. Functionally, it supports schema-based classification and extraction workflows, fine-tuning of LLMs, and multi-format input compatibility. From a technical perspective, it ensures modularity, cloud scalability, and seamless LLM integration. Economically, it is optimized for sub-cent processing costs per document. Finally, the platform is aligned with United Nations Sustainable Development Goals by promoting digital equity and process transparency, particularly in resource-constrained organizations.

4. Technical Description

The platform is built as a modular system composed of OCR, extraction, fine-tuning, and infrastructure modules. Documents are first digitized using AWS Textract or Mistral OCR, depending on layout complexity. Extraction is schema-driven, where each schema defines the structure, content type, and validation logic of the target data. LangChain is used to orchestrate LLM interactions, and OpenAI's fine-tuning API enables domain-specific improvements. The infrastructure, deployed on AWS, leverages event-driven pipelines for scalability and performance, incorporating services such as Lambda, S3, SQS, and DynamoDB for full automation and auditability (AWS, n.d.) (OpenAI, n.d.).

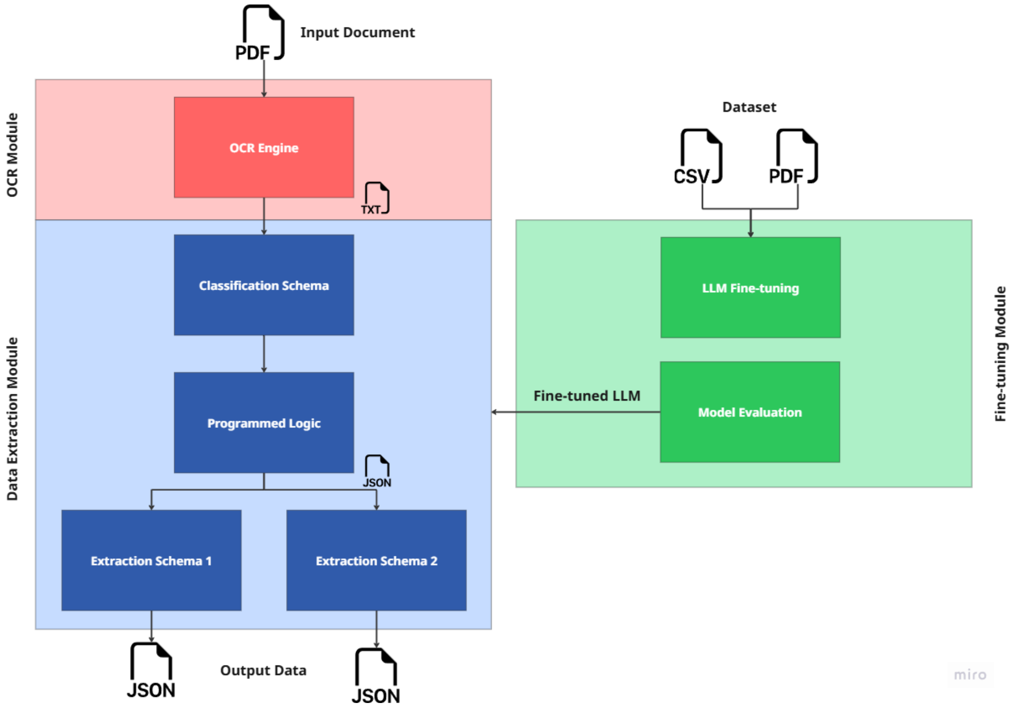


Figure 1.3 Document Intelligence System Modules

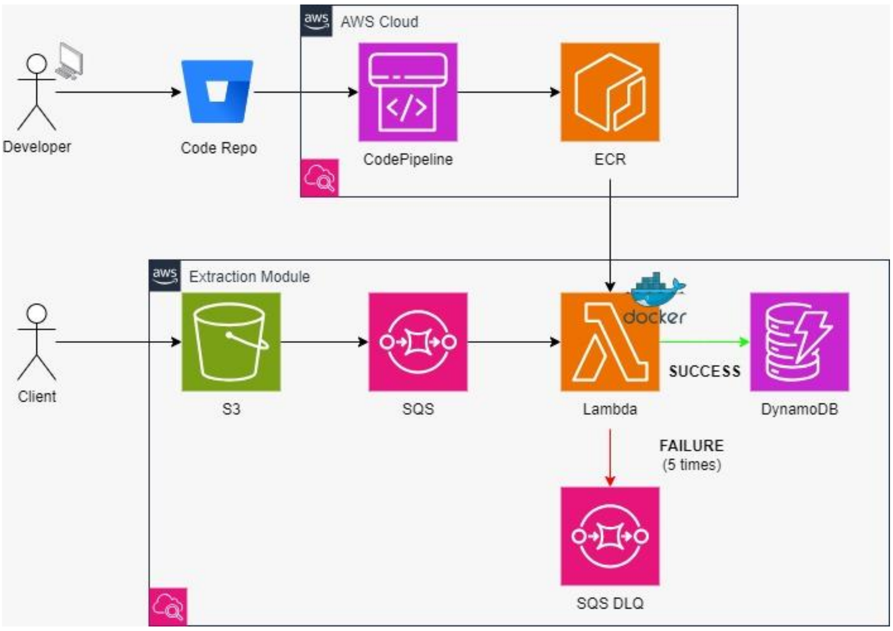


Figure 1.4 AWS-Based Document Processing Architecture

5. Practical Integrations in Enterprise Workflows

The system has been deployed in three distinct production settings to validate its versatility and robustness. In a legal firm managing mortgage foreclosures, the tool processed over 300 document types, achieving over 87% classification accuracy and 90% field-level precision. A multinational traffic fine processor applied the platform across five countries and multiple languages, reaching 91.3% extraction accuracy and reducing document handling times by half. Finally, within an internal platform (WorkOps), end users were able to define and execute custom schemas through a user interface, illustrating the system's accessibility and adaptability even in low-code environments.

6. Conclusions

While constructing a prototype document intelligence system is relatively straightforward with modern tools, building a production-grade solution demands significant effort in robustness, security, and usability. This project demonstrates that human-in-the-loop validation, auditability, and schema-driven configuration are not compromises, but essential design principles in enterprise AI. The platform's modular architecture, cost-aware design, and open fine-tuning capabilities enable its adoption across industries, laying the groundwork for scalable and explainable document automation.

7. References

AWS. (n.d.). AWS Products. Retrieved from https://aws.amazon.com/es/

Extend. (n.d.). Extend AI. Retrieved from https://www.extend.ai/

Invofox. (n.d.). Invofox. Retrieved from https://www.invofox.com/es

M-Files. (2019). M-Files. Retrieved from https://www.m-files.com/wp-content/uploads/2023/07/ebook-2019-intelligent-information-management-benchmark-en.pdf.pdf

Mistral. (n.d.). Mistral. Retrieved from https://mistral.ai/news/mistral-ocr

Nanonets. (n.d.). Nanonets. Retrieved from https://nanonets.com/

OpenAI. (n.d.). OpenAI API Docs - GPT Models. Retrieved 06 27, 2023, from https://platform.openai.com/docs/guides/gpt

Pydantic. (n.d.). Pydantic. Retrieved from https://docs.pydantic.dev/latest/

Reducto. (n.d.). Reducto AI. Retrieved from https://reducto.ai/

TableFlow. (n.d.). TableFlow. Retrieved from https://tableflow.com/

Índice de la memoria

Índice de figuras

Chapter 1. INTRODUCTION

The rapid growth of digital transformation has exposed a critical gap in how organizations process unstructured documents. From contracts and invoices to reports, forms, and memos, business-critical information is often embedded in formats that are difficult to classify, extract, or analyze at scale. Despite major advances in AI and natural language processing, most companies still rely on brittle rule-based pipelines or proprietary black-box models that fail to generalize across domains, require extensive manual setup, or incur high operational costs.

This Master's Thesis introduces a general-purpose document intelligence platform designed to address these limitations. The system enables users to automatically classify and extract structured data from any document type—without predefined templates, annotated datasets, or complex development cycles. Central to this solution is a schema-driven architecture: users can define the data they wish to extract, as well as optional classification categories, using plain-text descriptions formalized in Python schemas. These schemas serve as unified instruction sets for both classification and extraction, abstracting away the complexity of prompt engineering and making the system accessible, flexible, and reusable across domains.

The platform integrates state-of-the-art tools such as AWS Textract and Mistral OCR for layout-aware text extraction, alongside LLM backends like GPT-4o and GPT-4.1 for semantic understanding. A fine-tuning pipeline has also been developed to train domain-specific models for higher precision in specialized workflows. Thanks to its modular design and infrastructure on AWS, the tool supports low-cost, high-volume document processing while remaining scalable, explainable, and enterprise-ready.

The system has already been applied in diverse production environments—from mortgage legal processing and multilingual traffic fines to internal platforms—demonstrating high classification accuracy, robust field-level extraction, and reduced manual effort.

Chapter 2. MARKET ANALYSIS

After defining the goals and scope of this Master's Thesis, it is essential to determine whether a genuine market need exists for a general-purpose document-intelligence system and to understand how well existing products satisfy that need. This chapter therefore pursues three objectives: (1) quantify the demand for automated document classification and data extraction across multiple industries; (2) review representative competitors to identify capability gaps; and (3) compare the projected cost of processing documents with the proposed tool against alternative commercial offerings, thereby establishing an economic justification for further development.

2.1 BUSINESS NEED

In this section we examine why organizations require a flexible, general-purpose document-intelligence tool. First, we present the core problems faced by companies that handle large volumes of heterogeneous documents. Second, we explain how recent advances in artificial intelligence can address those problems.

2.1.1 IDENTIFYING THE PROBLEM

In today's data-driven economy, organizations of all sizes manage an increasingly vast volume of documents as part of their daily operations. These include contracts, invoices, purchase orders, delivery notes, internal reports, emails, technical datasheets, and legal documents. According to industry estimates, the average mid-sized company processes over 500,000 documents annually, with larger enterprises often exceeding several million per year—particularly in regulated industries like finance, insurance, and law.

This need is especially pressing in high-document-volume sectors:

- **Banking:** Large banks often handle millions of documents per day across their branches and corporate offices. This includes account applications, loan forms, transaction records, and regulatory reports. Even medium-sized banks frequently report manually processing thousands of documents per day (Emerj, 2019) (Artsyl, n.d.).

- **Insurance:** The insurance industry is among the most document-intensive sectors. Large insurance providers process tens of thousands of documents daily, including policy applications, claims, underwriting documents, and customer correspondence. In automated workflows, this number can exceed 400,000 documents per day (Newswire, 2021).
- **Legal:** Law firms also face significant document processing challenges. A single attorney may generate between 20,000 and 100,000 pages of documents per year through contracts, legal filings, client communication, and internal memos. At the firm level, this results in several million documents per year, even for mid-sized practices (Edmondson, 2018).

Despite the strategic importance of these documents, the vast majority remain underutilized. Unstructured documents are typically stored across disparate systems—shared drives, email inboxes, local folders, hard copies, legacy archives—creating fragmented, inconsistent, and unsearchable silos of information. This fragmentation leads to significant operational inefficiencies and hampers decision-making processes that rely on timely and accurate access to data.

Below, we outline the key challenges organizations face when dealing with large volumes of heterogeneous documents:

1. Lack of centralization and visualization

Most organizations lack a unified platform or system for managing and interacting with their document base. Documents are often stored in multiple, unconnected repositories—such as file servers, cloud drives, scanned archives, and even physical filing cabinets—which significantly limits centralized access. A study by M-Files indicates that 83% of employees have had to recreate documents they couldn't find, and 91% say their work is negatively affected by poor document management practices (M-Files, 2019).

This lack of centralization makes it difficult to maintain an overview of what information exists within the organization, where it is stored, and in what format. It also hinders the ability to apply analytics, controls, or security policies consistently. Without visibility and governance over document flows, valuable insights remain hidden, and compliance risks increase.

2. No value extracted from contracts

In many companies, documents are treated as static repositories of information—used once, archived, and rarely revisited unless a problem arises. This approach significantly limits the potential business value that can be extracted from document content.

Whether it's identifying pricing trends in supplier invoices, extracting KPIs from performance reports, or aggregating key clauses from legal documents, the structured data embedded in unstructured documents can offer critical business insights. However, without tools to automatically extract, standardize, and analyze this information, it remains locked away. As a result, documents become "dead" artifacts—retained for compliance, but offering little operational or strategic value.

The lack of document intelligence also impedes the development of data-driven processes. Without automatic data extraction, organizations rely on manual input, which is error-prone, time-consuming, and expensive.

3. No searchability or accessibility

The absence of a structured and intelligent document processing system results in severe limitations on searchability and accessibility. Employees often struggle to locate the right documents or extract the specific information they need—especially when dealing with multi-page scanned documents, multi-language files, or various file formats (PDFs, images, Word files, etc.).

Document analysis, encompassing classification and data extraction, is the foundational step toward solving this problem. By converting unstructured documents into structured formats, relevant fields and categories become machine-readable. Once indexed, documents can be filtered by type, date, client, amount, or any other extracted variable, making them immediately searchable and usable across departments and workflows.

Implementing such capabilities significantly accelerates decision-making, supports regulatory compliance (e.g., auditing), and improves productivity. Ultimately, searchable and structured document data enables a more agile, informed, and scalable organization.

2.1.2 HOW DOCUMENT INTELLIGENCE SOLVES THESE PROBLEMS

Once seen the problems that can arise from not having proper a contract management system in place, we can now list the numerous benefits of having a proper contract management system.

1. Increase productivity

One of the most immediate and measurable benefits of Document Intelligence is the substantial increase in productivity. By automating tasks such as document classification, information retrieval, and data entry, organizations drastically reduce the time spent manually processing files.

Employees no longer need to search through folders, open individual documents, or manually transcribe key data into systems of record. With intelligent indexing and metadata extraction, relevant documents can be retrieved in seconds based on structured queries (e.g., "invoices from supplier X in Q1 2024"). The automation of repetitive and error-prone tasks also reduces bottlenecks in operational workflows, enabling teams to focus on higher-value activities.

2. Reduce risks

Errors and oversights in document handling can have serious consequences. Missing a deadline for submitting a legal document, failing to detect a compliance clause in a contract, or using outdated pricing terms in a quotation can result in penalties, reputational damage, or financial losses.

Document Intelligence reduces these risks by ensuring that critical information is extracted accurately and on time. By automating the extraction of dates, obligations, terms, or compliance indicators, the system ensures that no key detail is overlooked. Moreover, it eliminates much of the human error associated with manual review, increasing the reliability of internal controls and audit readiness.

3. Value extraction from documents

Documents are not just records; they are rich repositories of strategic information. However, without structured access to their contents, much of their value remains untapped. Document Intelligence enables organizations to unlock that value by surfacing patterns, trends, and insights embedded in the textual and tabular content of files.

For example, companies can identify the most common types of customer claims from support tickets, analyze the frequency of specific clauses in legal agreements, or calculate average delivery times from scanned packing slips. By turning static documents into dynamic data assets, organizations create opportunities for optimization, revenue enhancement, and innovation.

4. Enhanced Decision-Making

Perhaps the most transformative benefit of Document Intelligence is its ability to empower smarter and faster decision-making. When organizations have structured access to the information inside their documents, they can monitor performance indicators, detect anomalies, and make data-driven choices with greater confidence.

Whether it's identifying discrepancies between purchase orders and delivery notes, benchmarking supplier performance, or proactively flagging renewal dates, the availability of timely and relevant data enhances strategic planning and execution. The intelligence extracted from documents can also feed into dashboards, forecasting models, and cross-departmental analytics—bridging the gap between raw data and actionable insight.

Summary

In summary, Document Intelligence addresses the core pain points associated with traditional document handling by improving access, accuracy, and analytical value. It transforms unstructured content into a structured asset, delivering operational efficiency, regulatory control, and business insight in a single unified framework.

2.2 COMPETITORS

As the adoption of AI in document processing accelerates, a new generation of companies is emerging to address the growing need for flexible, accurate, and scalable document understanding systems. Many of these startups—some of which are backed by Y Combinator—have developed platforms that allow organizations to extract and structure information from unstructured documents such as invoices, contracts, forms, reports, and ID documents.

In this section, we present five representative competitors that stand out for their innovation, technical depth, and configurability. These solutions are capable of performing automatic classification, data extraction, and layout understanding using OCR, LLMs, or a combination of both. The five vendors discussed below were chosen because they provide publicly documented APIs, publish transparent pricing, and actively market AI-based extraction.

2.2.1 SELECTED COMPETITORS

1. Extend

Extend is a modern document processing platform that uses a combination of OCR, vision models, and LLMs to automate extraction, classification, and document splitting. It is designed for enterprise-grade accuracy and supports complex layouts (e.g., tables, forms, handwriting). Extend includes a low-code UI for human-in-the-loop validation and continuous fine-tuning, making it highly customizable for diverse industries like finance, healthcare, and logistics. (Extend, n.d.)

2. TableFlow

TableFlow focuses on transforming PDFs, spreadsheets, and images into structured data. It allows users to define destination schemas—lists of expected fields and data types—which act as extraction targets. Using semantic AI and visual context analysis, it automatically maps and validates extracted content against the schema. It is especially effective for high-volume onboarding, financial reconciliation, and ERP data ingestion tasks. (TableFlow, n.d.)

3. Invofox

Invofox provides a developer-friendly API to extract structured data from financial and operational documents, such as invoices, receipts, bills of lading, and HR forms. Its key value proposition is rapid deployment without needing templates or human supervision. Through its "Custom Documents" module, clients can define which fields to extract for each document type, effectively enabling schema-based extraction across industries. (Invofox, n.d.)

4. Reducto

Reducto is a high-accuracy API platform that handles complex document layouts with a vision+LLM hybrid pipeline. It allows users to specify the fields they want to extract in natural language, and returns structured JSON accordingly. It excels in processing long and dense documents such as financial statements, legal contracts, and academic papers, and includes features like table extraction, chart understanding, and layout-aware parsing. (Reducto, n.d.)

5. Nanonets

Nanonets offers a no-code document automation platform with pre-trained and user-trainable models for extracting data from a wide variety of business documents. Its drag-

and-drop workflow builder allows teams to configure end-to-end pipelines—including document ingestion, classification, extraction, validation, and integration with external systems (e.g., ERP/CRM). It supports schema training via example documents and is widely used in finance, healthcare, and insurance. (Nanonets, n.d.)

2.2.2 Feature Comparison Table

Feature	***Extend***	***TableFlow***	***Invofox***	***Reducto***	***Nanonets***
Configurable Extraction (Schemas)	Yes	Yes	Yes	Yes	Yes
Document Classification	Yes	Yes	Yes	Yes	Yes
OCR + Layout Parsing	Advanced	Advanced	Moderate	Advanced	Moderate
LLM-Powered Extraction	Yes	Yes	Partial	Yes	Yes
Prebuilt Models	No	No	Yes	No	Yes
Custom Model Training	Yes	Yes	Yes	Not needed	Yes
Human-in-the-Loop Review	Yes	Yes	Optional	Optional	Yes
Multi-language Support	Yes	Yes	Yes	Yes	Yes
Deployment Options	Cloud + API	Cloud + API	API only	API + Private	Cloud + On-prem
Free Trial	Yes	Yes	Yes	Yes	Yes
Estimated Starting Cost	$300/mo	$500/mo	Custom	Custom	$0.30/page or $999/mo plan

2.2.3 CONCLUSION OF THE COMPETITOR ANALYSIS

The five vendors reviewed—Extend, TableFlow, Invofox, Reducto, and Nanonets—demonstrate that the market already delivers impressive point solutions for OCR, layout parsing, and LLM-driven extraction. However, when their capabilities are set side-by-side three clear gaps emerge:

- **Cost at Scale**: Public pricing remains at €0.01 per document once OCR + LLM usage is included, especially for documents that exceed one page or contain tables.
- **Schema Portability and Re-use**: Each platform keeps extraction logic in a proprietary project format. Moving a schema—or even a single field definition—from one vendor to another is effectively a rebuild. This undermines vendor independence and slows multi-tenant deployments.
- **Open Fine-Tuning Path:** Custom model training is either unavailable (Reducto) or gated behind enterprise-tier contracts (Extend, Nanonets). Organizations with domain-specific terminology therefore face high switching costs or must accept lower accuracy.

These gaps validate the design goals formulated in Chapter 1 and translated into technical and economic requirements in Chapter 3: a solution that achieves sub-cent-per-document costs, represents extraction logic as LLM-agnostic Pydantic schemas, and exposes an accessible fine-tuning pipeline. The remainder of the thesis shows how the proposed architecture fills precisely the cost, portability, and customisation spaces that current market offerings leave open.

Chapter 3. REQUIREMENTS AND DESIGN PRINCIPLES

After analyzing the business needs and defining the value proposition, this section outlines the functional and non-functional requirements that the developed tool must satisfy. The requirements are divided into three categories: **functional, technical, economic, and social**.

3.1 FUNCTIONAL REQUIREMENTS

- **Schema-based configurability**
  The tool must allow the user to define what data they want to extract or which categories they want to classify, through intuitive and lightweight extraction schemas. These schemas must be easy to define and adaptable to different document types (e.g., invoices, contracts, reports, etc.).
- **Dual-purpose capability**
  The tool must be capable of both classifying documents (e.g., identifying type, department, purpose) and extracting structured information from them. The same schema definition must enable both tasks if needed.
- **Fine-tuning support**
  The platform must support fine-tuning of large language models (LLMs) for improved accuracy in specific domains or document types. This functionality must be accessible and manageable through a UI or API.
- **Multi-format input**
  It must accept documents in various formats, including PDFs (native and scanned), Word documents, images (JPG/PNG), and email attachments.

3.2 TECHNICAL (NON-FUNCTIONAL) REQUIREMENTS

- **Cloud-native scalability**

  The solution must be designed for scalable cloud deployment, with the ability to parallelize processing across large volumes of documents efficiently.
- **LLM modularity**

  The system must be designed to easily plug in different LLM backends (e.g., GPT, Claude, Gemini) depending on performance, cost, and security considerations.
- **Human-in-the-loop readiness**

  While aiming for full automation, the architecture must be ready to integrate human validation steps when required.
- **Cross-platform compatibility**

  Although cloud-first, local development and testing must remain compatible with Windows, macOS, and Linux environments.

3.3 ECONOMIC REQUIREMENTS

The average cost per document processed (including OCR and LLM inference) should remain significantly below typical market prices. The target is to keep it under €0.01 per document when using cost-efficient models like GPT-4o mini and AWS Textract.

3.4 SOCIAL REQUIREMENTS

The tool must align with the United Nations' Sustainable Development Goals (SDGs), particularly by democratizing access to AI document processing in small and medium-sized enterprises (SMEs), and by promoting transparency, digitalization, and operational efficiency.

Chapter 4. TECHNICAL DESCRIPTION

4.1 USED TECHNOLOGIES

The technological stack employed in this project was selected to ensure flexibility, modularity, and high performance in large-scale document processing. The solution integrates open-source libraries, cloud-native services, and advanced AI models.

Python

Python serves as the core development language due to its extensive ecosystem for AI and web development. Key libraries include pandas, pydantic, FastAPI, and uvicorn, which support robust data handling, schema validation, and API orchestration (Pydantic, n.d.) (FastAPI, n.d.).

LangChain

LangChain is used to manage LLM workflows across different providers. It abstracts prompt construction, memory management, and structured output parsing, enabling schema-based document extraction and agent-driven logic. (LangChain, n.d.)

OpenAI API & Fine-Tuning

The platform utilizes OpenAI's API for tasks such as classification, field extraction, and table parsing, primarily using GPT-4 and GPT-4o. Fine-tuning capabilities are leveraged to train models on domain-specific data, improving precision. OpenAI's monitoring tools allow real-time tracking of training progress and model performance.

Mistral OCR

To handle complex layouts and handwritten inputs, Mistral's OCR models are integrated as an alternative to standard OCR. These models provide high accuracy for irregular or low-quality documents, albeit at a higher computational cost. (Mistral, n.d.)

AWS Services and Infrastructure

The platform is deployed using a modular, event-driven architecture on Amazon Web Services (AWS), divided into a deployment pipeline and a document extraction pipeline (AWS, n.d.). Key services include:

- AWS CodePipeline, ECR, and Lambda for continuous deployment of containerized application components
- Amazon S3 and SQS to manage file ingestion and job queuing
- DynamoDB for structured storage of extracted metadata and schema outputs
- AWS Lambda (containerized) for executing OCR, classification, and extraction logic at scale

Within this pipeline, **AWS Textract** is integrated as the primary OCR engine. It delivers accurate extraction of key-value pairs and table structures from scanned documents and natively fits into the Lambda-based architecture. Its pricing and output quality make it well-suited for high-throughput, production-grade document processing.

4.2 System Modules and Analysis Pipeline

The architecture of the developed tool is designed to be modular, extensible, and cloud-ready. It is composed of four main components: the OCR Module, the Data Extraction Module, the Fine-tuning Module, and the Infrastructure Module. Each component operates independently and communicates through well-defined interfaces to enable end-to-end document understanding workflows.

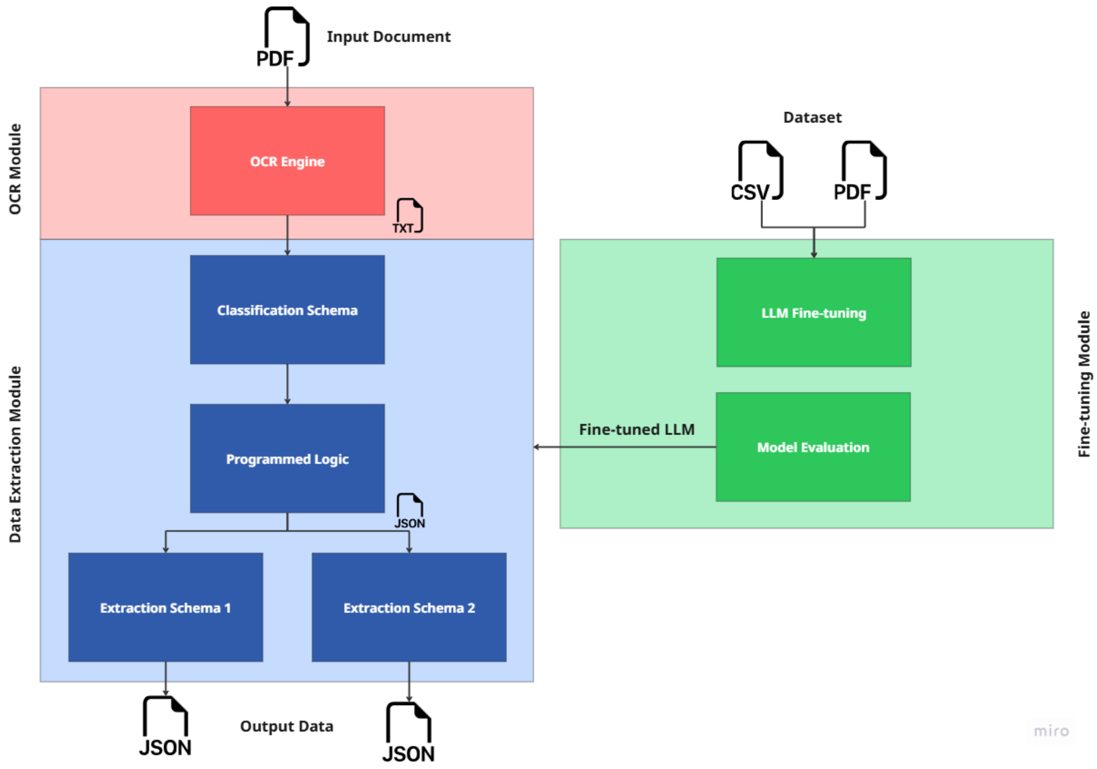


Figure 4.1 System modules diagram

4.2.1 OCR MODULE

The OCR Module is responsible for converting document files (PDF, Word) into structured and machine-readable text. It is designed to operate independently from downstream processes and supports two OCR engines:

- **AWS Textract:** Provides accurate extraction of key-value pairs and table structures from printed documents, and is well-suited for structured forms and standard layouts.
- **Mistral OCR**: Optimized for high-accuracy recognition in documents with complex layouts, handwriting, or low scan quality.

OCR is a well-established problem space with intense commercial competition and rapid innovation. For this reason, the system deliberately integrates third-party OCR services rather than developing a proprietary engine. This design choice allows the platform to leverage state-of-the-art accuracy without incurring the high R&D and infrastructure costs required to build and maintain an in-house OCR solution.

The choice between Textract and Mistral is not hardcoded—it is made dynamically at the schema level. This allows each schema to specify the preferred OCR engine based on its performance requirements, cost sensitivity, or data handling constraints. For example:

- Use Mistral OCR when dealing with messy scans, non-standard layouts, or when precision is critical.
- Use AWS Textract for standard business documents or when data privacy policies prohibit sending documents to third-party OCR providers outside AWS infrastructure.

This flexible configuration ensures that the system can adapt to diverse business needs while balancing trade-offs between cost, accuracy, and privacy.

4.2.2 DATA EXTRACTION MODULE

This is the central component of the system, responsible for both document classification and field-level data extraction. The module is designed to be fully configurable through lightweight user-defined schemas and supports extensible logic for managing complex workflows via ensemble functions.

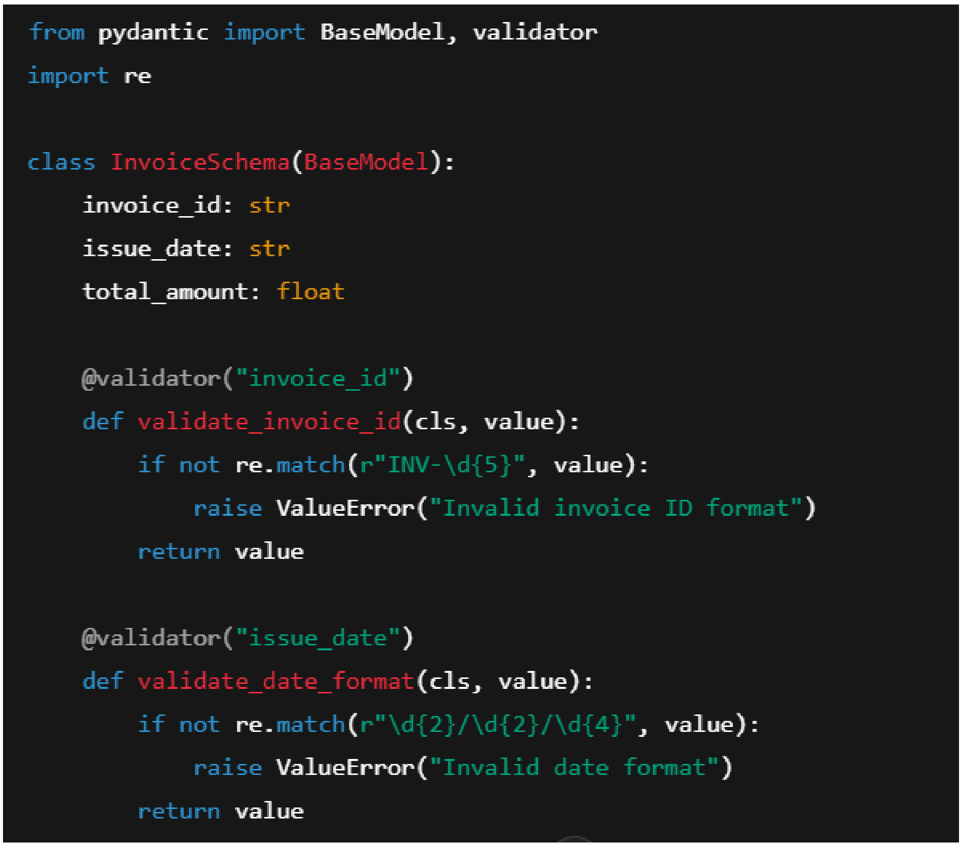
4.2.2.1 Extraction Schemas

Schemas are defined as Python classes using the Pydantic library (Pydantic, n.d.). Each schema describes a set of fields to extract, along with their metadata, data types, and validation constraints. Unlike traditional rule-based extractors, these schemas are declarative: they represent the structure of the output expected from a large language model (LLM).

A schema typically includes:

- **Field name:** the target field to extract (e.g., invoice_date)

- **Description:** semantic context to guide the LLM
- **Data type:** enforced via pydantic (e.g., str, float, datetime)
- **Validation rules:** optional functions for custom checks



```python
from pydantic import BaseModel, validator
import re

class InvoiceSchema(BaseModel):
    invoice_id: str
    issue_date: str
    total_amount: float

    @validator("invoice_id")
    def validate_invoice_id(cls, value):
        if not re.match(r"INV-\d{5}", value):
            raise ValueError("Invalid invoice ID format")
        return value

    @validator("issue_date")
    def validate_date_format(cls, value):
        if not re.match(r"\d{2}/\d{2}/\d{4}", value):
            raise ValueError("Invalid date format")
        return value
```

Figure 4.2 Pydantic schema example

In addition, the schema contains global configuration parameters, the most important one being:

- **OCR engine** to use (textract or mistral)
- **LLM model** to use (gpt-4o, gpt-4o-mini, fine-tuned LLM, etc.)

This structure enables fully modular and reproducible extraction tasks, tailored per document type or business domain.

Schema Creation Criteria

Schemas should be created whenever a new type of document needs to be extracted or a new classification category must be introduced. That means:

- One schema per **document type** (e.g., invoice, contract, delivery note)
- One schema per **classification use case** (e.g., detecting document purpose or department)

In practice, any time the structure or semantic intent of a document differs in a way that affects what information needs to be extracted—or how that information is organized—a new schema is warranted. For recurring or similar formats, existing schemas can be reused or extended to reduce duplication.

This schema-centric approach ensures clarity, modularity, and ease of maintenance as the system scales across document types and use cases.

4.2.2.2 Extraction Engine

The core mechanism of data extraction engine is built on top of LangChain's structured_output() functionality (Pydantic, n.d.). The LLM is prompted to fill in the fields defined in the schema class, and the returned JSON is automatically validated and parsed into a strongly typed pydantic object.

The extraction flow works as follows:

1. The document is first parsed by the selected OCR engine (Textract or Mistral).
2. A dynamic prompt is generated using the schema metadata.
3. The selected LLM processes the prompt and returns a JSON-compatible response.
4. LangChain parses the response and maps it to the schema fields, applying all validation checks.
5. The final result is either accepted or flagged for review based on validation outcomes.

This approach abstracts away prompt engineering and allows schema definitions to act as instruction sets for the LLM.

Validation Strategies

Validation logic can be injected at the schema level through simple Python functions. The system supports a range of validation types:

- Type validation (e.g., date, float, string)
- Regex pattern checks (e.g., invoice formats, ISO codes)
- Range enforcement (e.g., amounts must be > 0)
- Set membership (e.g., country must be in a predefined list)
- Cross-field checks (e.g., start_date < end_date)
- Custom domain-specific logic

This layered validation ensures that extracted data is reliable before it is saved or passed to downstream analytics systems.

4.2.2.3 Ensemble Workflows

For more complex workflows, the system introduces the concept of an ensemble—a function that orchestrates the execution of multiple schemas in sequence or conditionally.

Ensembles allow you to:

- Classify documents (e.g., invoice, delivery note, contract) using a generic schema.
- Route documents to specialized extraction schemas based on classification.
- Chain schemas to perform hierarchical or nested extraction (e.g., header fields + product tables).
- Apply business logic conditionally based on extracted values.

Unlike other systems, the distinction between classification and extraction schemas does not exist at the schema level. The ensemble logic decides how each schema is used, keeping schema design decoupled and reusable.

4.2.3 FINE-TUNING MODULE

This module enables the training of custom LLMs tailored to specific use cases. It leverages OpenAI's fine-tuning API to optimize model performance on domain-specific examples.

Capabilities include:

- Preparing prompt/response training pairs from annotated documents
- Launching and monitoring fine-tuning jobs on OpenAI infrastructure
- Visualizing training performance over time (e.g., loss curves)
- Plug-and-play integration with the rest of the system

Fine-tuned models are particularly useful in high-stakes environments (e.g., legal, healthcare) where generic models may fall short in precision.

Fine-Tuning Configuration and Prompt Automation

The fine-tuning module is designed to streamline the adaptation of base LLMs (primarily GPT-4o-mini) to domain-specific extraction tasks. The choice for GPT-4o-mini and GPT-4.1-mini is driven by its excellent balance between latency, accuracy, and cost, making them the default models for most fine-tuning jobs. However, the module remains compatible with other OpenAI model variants if additional performance is required.

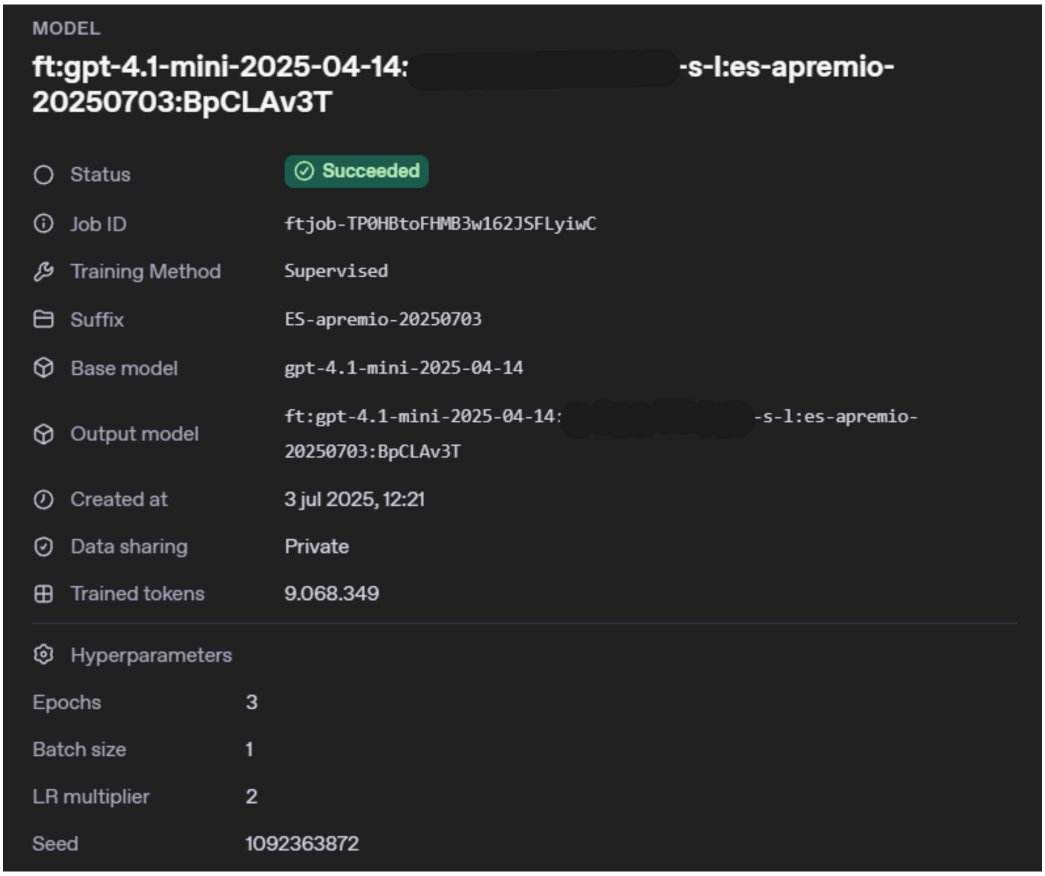
To fine-tune a model, a set of prompt–completion pairs must be provided. In this platform, those pairs are automatically generated using predefined Pydantic extraction schemas. The system imports the target schema and uses it to construct structured prompts and expected outputs from annotated training documents. This abstraction allows developers to fine-tune models without manually designing each prompt, improving reproducibility and accelerating the training pipeline.

In addition, several hyperparameters must be defined prior to launching a fine-tuning job. These include:

- Number of training epochs (typically 3–5),

- Batch size (depends on dataset size),
- Learning rate multiplier, and
- Validation split ratio.

These parameters directly influence convergence speed and generalization. The optimal configuration often depends on dataset quality and task complexity. The module exposes these settings through a configuration file and integrates sanity checks to prevent invalid runs.



MODEL

ft:gpt-4.1-mini-2025-04-14:[redacted]-s-l:es-apremio-20250703:BpCLAv3T

Status	Succeeded
Job ID	ftjob-TP0HBtoFHMB3w162JSFLyiwC
Training Method	Supervised
Suffix	ES-apremio-20250703
Base model	gpt-4.1-mini-2025-04-14
Output model	ft:gpt-4.1-mini-2025-04-14:[redacted]-s-l:es-apremio-20250703:BpCLAv3T
Created at	3 jul 2025, 12:21
Data sharing	Private
Trained tokens	9.068.349

Hyperparameters

Epochs	3
Batch size	1
LR multiplier	2
Seed	1092363872

Figure 4.3 OpenAI API Platform: fine-tuning hyperparameters

4.2.3.1 Evaluation Module

One of the fundamental challenges associated with the use of large language models (LLMs) lies in their inherent lack of interpretability. These models function as non-transparent systems in which the mechanisms that produce specific outputs are not readily accessible or understandable to the user. This opacity poses a particular problem in structured information extraction tasks, where the system is expected to extract precise data points from natural language content.

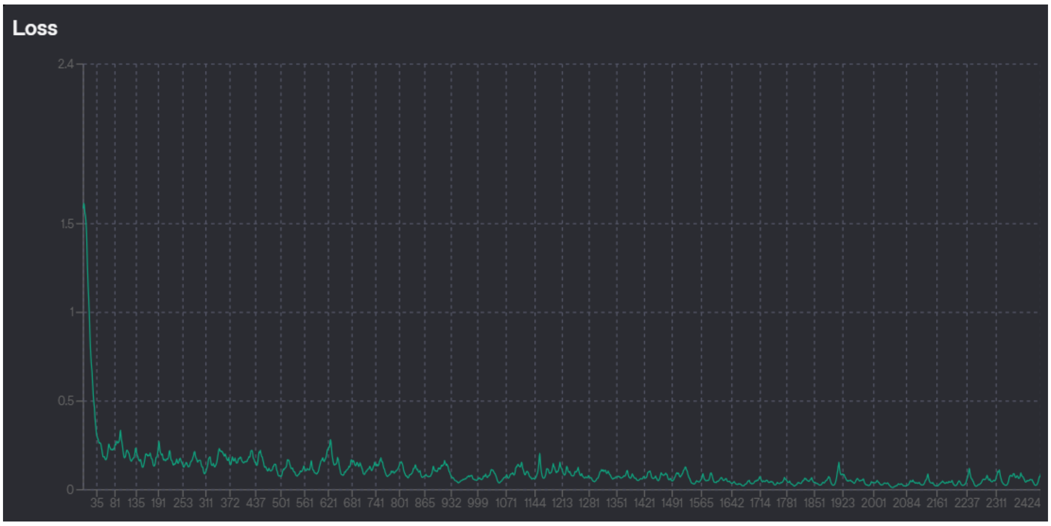


Figure 4.4 OpenAI API Platform: fine-tuning loss metrics

In such scenarios, model errors are frequently subtle and not easily attributable to a specific failure. For instance:

- A date may be returned in an incorrect format, despite representing a correct temporal reference.
- A numeric field may contain an incorrect value, possibly extracted from an unrelated section of the document.
- A document may be classified under an inappropriate category due to semantic ambiguity or prompt misalignment.

These issues cannot be effectively diagnosed by merely inspecting the model's output, and they complicate both the development and the maintenance of reliable document intelligence systems. To address this limitation, a dedicated Evaluation Module has been developed. This module is designed to systematically assess the performance of both base and fine-tuned models, particularly with respect to their ability to produce accurate and structured outputs as defined by the corresponding extraction schema.

Objectives and Scope

The Evaluation Module serves as an internal benchmarking framework with the following primary objectives:

- To quantify extraction performance at the field level, identifying both correct and incorrect outputs for individual schema fields.
- To detect systematic failure patterns, such as recurring omissions, format inconsistencies, or domain-specific misinterpretations.
- To support comparative evaluation between model variants (e.g., base vs. fine-tuned models) using standardized metrics.
- To enable regression monitoring across schema or model versions, thereby supporting long-term system robustness and traceability.

These objectives are critical in production environments where automated extraction must meet strict quality thresholds, particularly in domains with regulatory or contractual constraints.

4.3 INFRASTRUCTURE (AWS)

The developed system is deployed in a cloud-native architecture using Amazon Web Services (AWS), with a focus on scalability, modularity, and automated lifecycle management. All components are provisioned and managed using Infrastructure as Code (IaC) through Terraform, which ensures consistency, reproducibility, and environment portability (AWS, n.d.).

The infrastructure is divided into two main functional domains: the deployment pipeline and the document extraction pipeline.

4.3.1 DEPLOYMENT PIPELINE

The deployment process is designed to automate the integration and delivery of application components, following continuous deployment (CD) practices. This pipeline is primarily used by developers and includes the following steps:

- **Code Repository:** Source code is maintained in a Git-based repository, which serves as the version-controlled entry point to the deployment workflow.
- **CodePipeline:** AWS CodePipeline automates the build and deployment stages. Upon detecting a new commit, it triggers a pipeline that compiles the application logic and packages it for containerization.
- **Elastic Container Registry (ECR):** The containerized code is stored in AWS ECR, from where it can be pulled by compute services such as AWS Lambda for execution. This approach decouples the runtime environment from the build system and ensures image immutability (AWS, n.d.).

This deployment pipeline enables safe and controlled rollout of updates, facilitating quick iterations while maintaining operational stability.

4.3.2 DOCUMENT ANALYSIS PIPELINE

The document analysis pipeline is responsible for processing user-submitted documents through the OCR and data extraction modules. It is triggered by end-user interactions and is composed of the following components:

- **Amazon S3:** Clients upload input documents (e.g., PDFs, Word files) to a dedicated S3 bucket. This storage layer ensures durability and serves as the initial trigger for processing.
- **Amazon SQS:** Once a new file is uploaded, an event is published to an Amazon Simple Queue Service (SQS) queue. This decouples the ingestion layer from the compute layer and allows for horizontal scaling of workers.
- **AWS Lambda:** The core processing logic (OCR, extraction, validation) is executed inside containerized AWS Lambda functions. These functions pull jobs from the

SQS queue, execute the configured extraction pipeline, and write the outputs to persistent storage. Lambda is used in conjunction with Docker to encapsulate all necessary dependencies and runtime environments.

- **Amazon DynamoDB:** Processed data and metadata (e.g., schema used, model applied, validation status) are stored in DynamoDB for traceability, auditability, and fast querying (AWS, n.d.).

This architecture enables both batch and real-time document processing while ensuring reliability and fault tolerance.

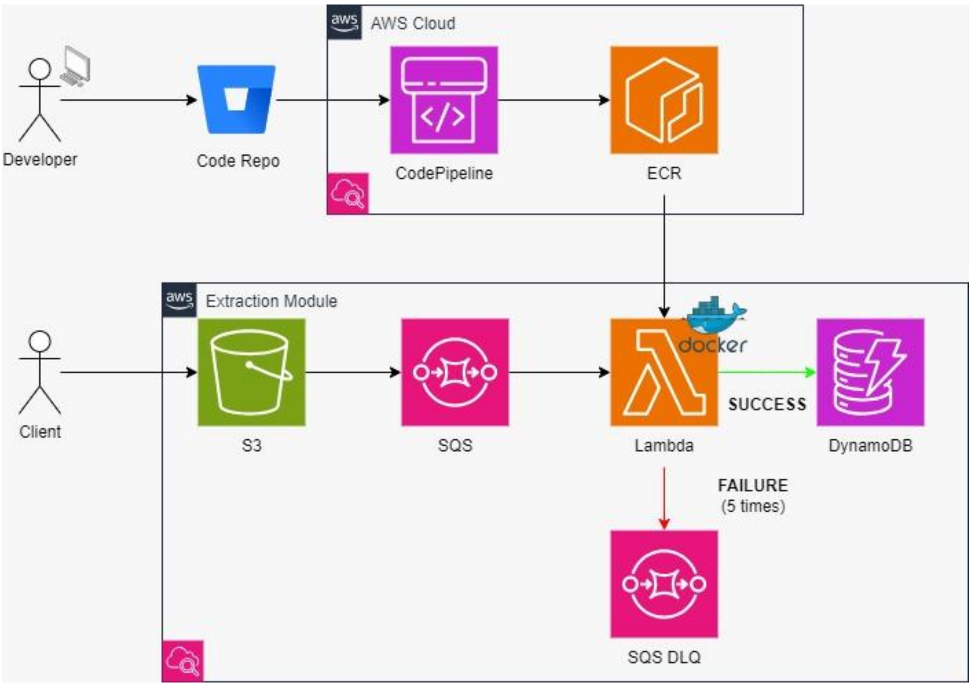


Figure 4.5 AWS architecture diagram

This modular and event-driven infrastructure ensures that the system can scale dynamically based on workload, provides clear operational boundaries between development and production, and supports fault isolation and recovery through established cloud patterns. By leveraging managed services and infrastructure automation, the system achieves a high degree of availability, maintainability, and cost-efficiency.

Chapter 5. PRACTICAL INTEGRATIONS IN ENTERPRISE WORKFLOWS

As organizations continue to digitize operational processes, there is increasing demand for scalable and intelligent document processing systems that can adapt to heterogeneous workflows, formats, and business domains. The solution developed in this thesis has been successfully integrated into several real-world enterprise environments. Each case demonstrates the tool's flexibility, configurability, and ability to handle complex, high-volume document streams across sectors.

This chapter presents three distinct deployment scenarios, illustrating how the document intelligence platform has been tailored to address domain-specific challenges using schema-driven extraction, LLM orchestration, and fine-tuned models. These examples showcase not only the technical robustness of the tool but also its practical applicability in high-stakes, production environments.

5.1 LEGAL DOCUMENT MANAGEMENT – MORTGAGE FORECLOSURE PROCEEDINGS

Client Profile: A company specialized in legal document processing for mortgage foreclosure proceedings, handling large volumes of administrative and procedural documents.

The company needed to automate the classification and data extraction from a heterogeneous set of legal documents related to foreclosure proceedings. These documents originated from multiple sources, including email attachments and SFTP repositories, and varied significantly in structure and format.

The total document corpus included more than 300 distinct document types, ranging from property title reports (notas simples), loan agreements, and court-issued procedural notices, to judicial decisions and enforcement orders.

System Configuration:

- **Schemas:** Over 20 customized schemas were developed to extract key legal entities (e.g., court reference numbers, borrower information, dates of judgment, enforcement status).
- **Ensemble:** A two-stage ensemble was applied:
  - Stage 1: Classification schema to identify the document type from over 50 critical categories, from the 300 available.
  - Stage 2: Document-specific extraction schema tailored to the identified class.
- **Model:** A model was fine-tuned for classification and 20 models for data extraction.

Results:

- Overall classification accuracy: 87%
- Average field-level extraction accuracy: +90% (for critical fields)
- Reduction in manual review effort: approx. 70%

The integration significantly accelerated document triage and data entry workflows, improving both compliance and operational efficiency.

5.2 TRAFFIC FINE PROCESSING – MULTINATIONAL ENFORCEMENT ENTITY

Client Profile: A private-sector company responsible for the management and administrative processing of traffic fines across multiple European countries.

The company receives traffic violation notices from over 18,000 issuing authorities across Spain, Italy, Portugal, France, and Germany. These documents vary by language, layout, format, and jurisdiction. The challenge was to standardize and extract key data fields—such as license plate, violation date, issuing body, and fine amount—across a multilingual and structurally inconsistent document base.

System Configuration:

- **OCR Engine:** Textract was used for clean, printed PDFs; fallback to Mistral OCR was enabled for low-quality scans.
- **Schemas:** A generalized multilingual schema was developed with language-specific field examples and formatting rules. Specific attention was paid to regional date formats and fine codes.
- **Ensemble:** One multilingual classification schema was used to identify the issuing authority or document layout, followed by conditionally applied extraction schemas per country.

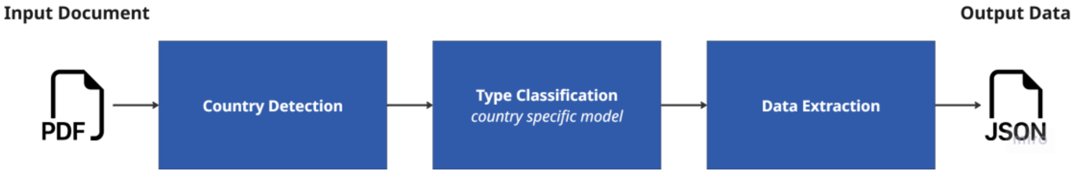


Figure 5.1 Ensemble of the traffic fine use case

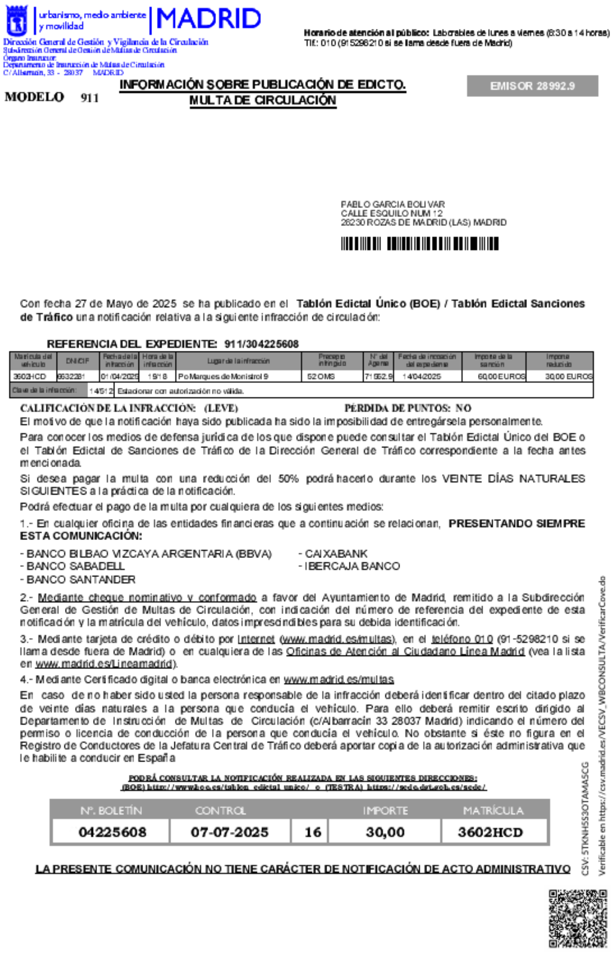
- **Model:** A total of 13 models have been fine-tuned so far; for type classification and data extraction for a given type.

Results:

- Average field-level extraction accuracy: 91.3%
- Daily processing volume: 2,000+ documents/day (60,000 documents/month)
- Document processing speed: x2 times faster than before.

This deployment enabled full automation of the ingestion pipeline, integrating seamlessly with the client's case management system and allowing for near real-time processing of incoming fines.

Below, is an example of a document and it data extracted.



urbanismo, medio ambiente y movilidad | MADRID
Dirección General de Gestión y Vigilancia de la Circulación
Subdirección General de Gestión de Multas de Circulación
Órgano Instructor:
Departamento de Instrucción de Multas de Circulación
C/ Albarracín, 33 - 28037 MADRID

Horario de atención al público: Laborables de lunes a viernes (8:30 a 14 horas)
Tlf.: 010 (915298210 si se llama desde fuera de Madrid)

MODELO 911

INFORMACIÓN SOBRE PUBLICACIÓN DE EDICTO.
MULTA DE CIRCULACIÓN

EMISOR 28992.9

PABLO GARCIA BOLIVAR
CALLE ESQUILO NUM 12
26230 ROZAS DE MADRID (LAS) MADRID

Con fecha 27 de Mayo de 2025 se ha publicado en el **Tablón Edictal Único (BOE) / Tablón Edictal Sanciones de Tráfico** una notificación relativa a la siguiente infracción de circulación:

REFERENCIA DEL EXPEDIENTE: 911/304225608

Matrícula del vehículo	DNI/CIF	Fecha de la infracción	Hora de la infracción	Lugar de la infracción	Precepto infringido	Nº del Agente	Fecha de incoación del expediente	Importe de la sanción	Importe reducido
3602HCD	6632281	01/04/2025	19:18	Po Marques de Monistrol 9	52 OMS	71552.9	14/04/2025	60,00 EUROS	30,00 EUROS
Clave de la infracción:	14/512	Estacionar con autorización no válida.							

CALIFICACIÓN DE LA INFRACCIÓN: (LEVE) **PÉRDIDA DE PUNTOS: NO**

El motivo de que la notificación haya sido publicada ha sido la imposibilidad de entregársela personalmente.

Para conocer los medios de defensa jurídica de los que dispone puede consultar el Tablón Edictal Único del BOE o el Tablón Edictal de Sanciones de Tráfico de la Dirección General de Tráfico correspondiente a la fecha antes mencionada.

Si desea pagar la multa con una reducción del 50% podrá hacerlo durante los VEINTE DÍAS NATURALES SIGUIENTES a la práctica de la notificación.

Podrá efectuar el pago de la multa por cualquiera de los siguientes medios:

1.- En cualquier oficina de las entidades financieras que a continuación se relacionan, **PRESENTANDO SIEMPRE ESTA COMUNICACIÓN:**

- BANCO BILBAO VIZCAYA ARGENTARIA (BBVA)
- BANCO SABADELL
- BANCO SANTANDER
- CAIXABANK
- IBERCAJA BANCO

2.- Mediante cheque nominativo y conformado a favor del Ayuntamiento de Madrid, remitido a la Subdirección General de Gestión de Multas de Circulación, con indicación del número de referencia del expediente de esta notificación y la matrícula del vehículo, datos imprescindibles para su debida identificación.

3.- Mediante tarjeta de crédito o débito por Internet (www.madrid.es/multas), en el teléfono 010 (91-5298210 si se llama desde fuera de Madrid) o en cualquiera de las Oficinas de Atención al Ciudadano Línea Madrid (vea la lista en www.madrid.es/Lineamadrid).

4.- Mediante Certificado digital o banca electrónica en www.madrid.es/multas

En caso de no haber sido usted la persona responsable de la infracción deberá identificar dentro del citado plazo de veinte días naturales a la persona que conducía el vehículo. Para ello deberá remitir escrito dirigido al Departamento de Instrucción de Multas de Circulación (c/Albarracín 33 28037 Madrid) indicando el número del permiso o licencia de conducción de la persona que conducía el vehículo. No obstante si éste no figura en el Registro de Conductores de la Jefatura Central de Tráfico deberá aportar copia de la autorización administrativa que le habilite a conducir en España

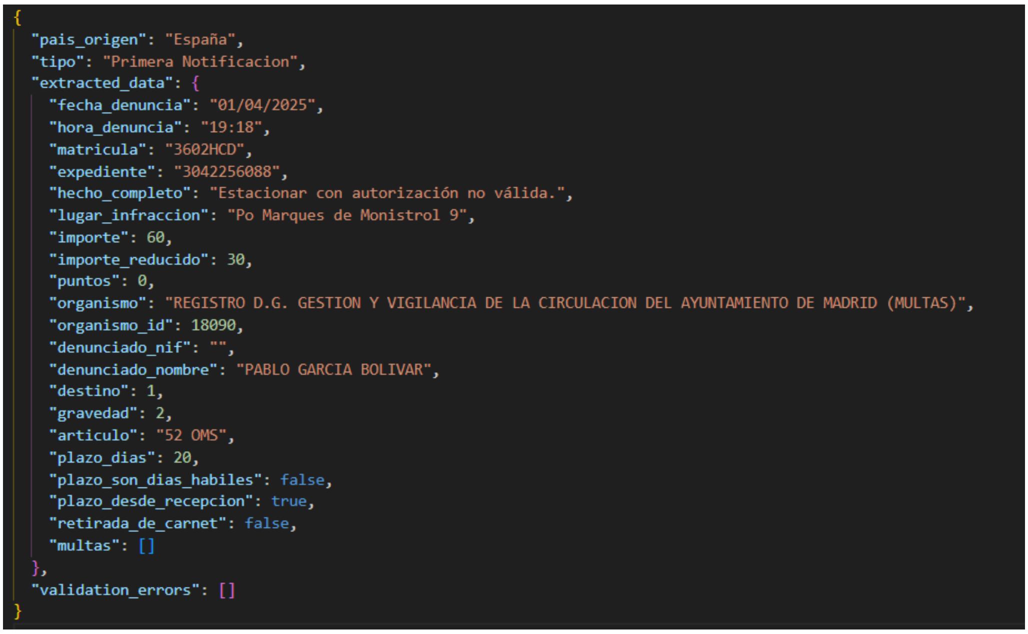
PODRÁ CONSULTAR LA NOTIFICACIÓN REALIZADA EN LAS SIGUIENTES DIRECCIONES:
(BOE) http://www.boe.es/tablon_edictal_unico/ o (TESTRA) https://sede.dgt.gob.es/sede/

Nº. BOLETÍN	CONTROL		IMPORTE	MATRÍCULA
04225608	07-07-2025	16	30,00	3602HCD

LA PRESENTE COMUNICACIÓN NO TIENE CARÁCTER DE NOTIFICACIÓN DE ACTO ADMINISTRATIVO

CSV: 5TKNHS53OTAMA5CG
Verificable en https://csv.madrid.es/VECSV_WBCONSULTA/VerificarCove.do

Figure 5.2 Traffic fine example



```
{
  "pais_origen": "España",
  "tipo": "Primera Notificacion",
  "extracted_data": {
    "fecha_denuncia": "01/04/2025",
    "hora_denuncia": "19:18",
    "matricula": "3602HCD",
    "expediente": "3042256088",
    "hecho_completo": "Estacionar con autorización no válida.",
    "lugar_infraccion": "Po Marques de Monistrol 9",
    "importe": 60,
    "importe_reducido": 30,
    "puntos": 0,
    "organismo": "REGISTRO D.G. GESTION Y VIGILANCIA DE LA CIRCULACION DEL AYUNTAMIENTO DE MADRID (MULTAS)",
    "organismo_id": 18090,
    "denunciado_nif": "",
    "denunciado_nombre": "PABLO GARCIA BOLIVAR",
    "destino": 1,
    "gravedad": 2,
    "articulo": "52 OMS",
    "plazo_dias": 20,
    "plazo_son_dias_habiles": false,
    "plazo_desde_recepcion": true,
    "retirada_de_carnet": false,
    "multas": []
  },
  "validation_errors": []
}
```

Figure 5.3 Extracted data example

5.3 *INTERNAL PLATFORM – WORKOPS (CUSTOM SCHEMA CONFIGURATION)*

Client Profile: Galileo Studio (internal project)

As part of an internal R&D initiative, the document extraction platform was integrated into WorkOps, a document-based workflow automation tool developed in-house. The goal of this integration was to provide end users with the ability to create and deploy their own extraction schemas via a user interface, without requiring programming knowledge.

System Configuration:

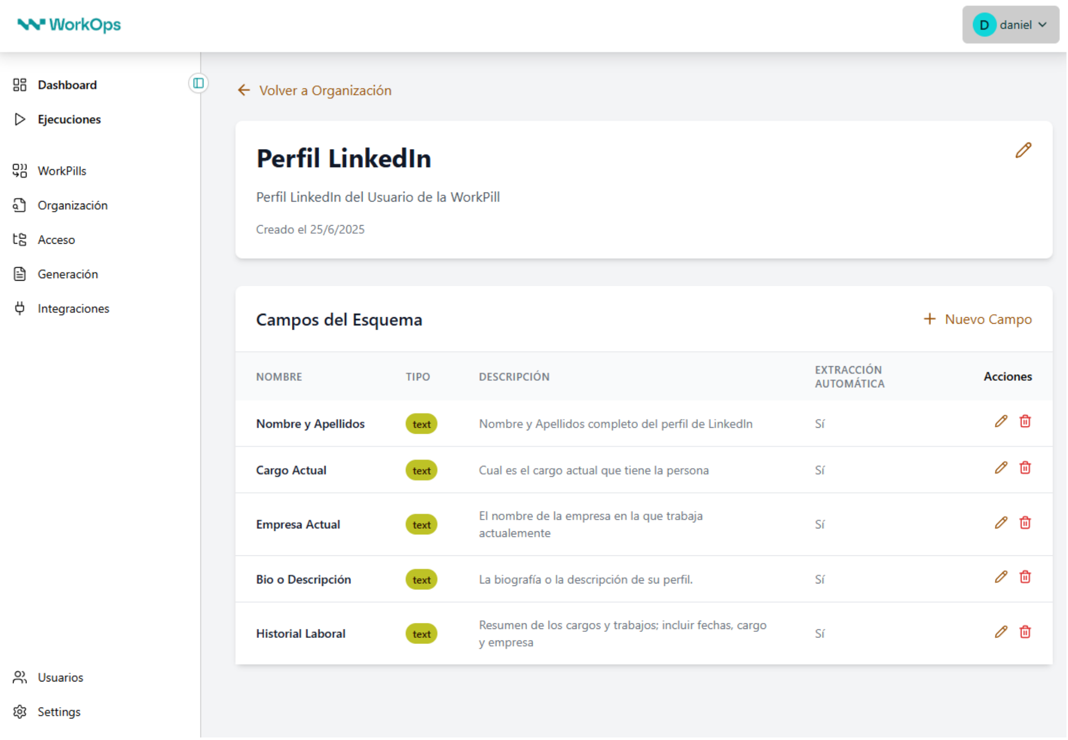
- **Schemas:** Dynamically defined via UI and stored in DynamoDB. Users can select fields, define types, examples, and format constraints.
- **Ensemble:** Logic automatically generated based on user inputs—classification and extraction schemas are dynamically composed and executed accordingly.

- **Model:** GPT-4o-mini was selected to balance cost and latency, as this use case involves frequent low-volume requests.

Results:

- Time to deploy new schema: under 2 minutes
- Schema adoption rate among pilot users: 80%
- Error rate in UI-defined extractions: <7% (acceptable given no fine-tuning and user-driven schema quality)

This deployment demonstrates the versatility of the platform when exposed as a low-code service, supporting highly flexible use cases without requiring deep technical integration or custom model training.



Perfil LinkedIn

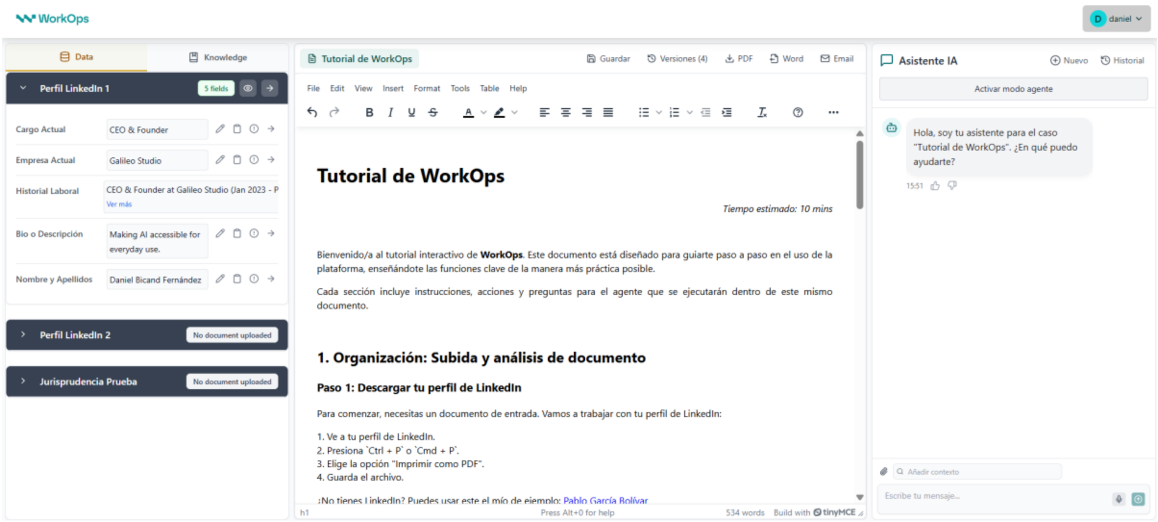
Perfil LinkedIn del Usuario de la WorkPill

Creado el 25/6/2025

Campos del Esquema

NOMBRE	TIPO	DESCRIPCIÓN	EXTRACCIÓN AUTOMÁTICA	Acciones
Nombre y Apellidos	text	Nombre y Apellidos completo del perfil de LinkedIn	Sí	
Cargo Actual	text	Cual es el cargo actual que tiene la persona	Sí	
Empresa Actual	text	El nombre de la empresa en la que trabaja actualemente	Sí	
Bio o Descripción	text	La biografía o la descripción de su perfil.	Sí	
Historial Laboral	text	Resumen de los cargos y trabajos; incluir fechas, cargo y empresa	Sí	

Figure 5.4 WorkOps platform: schema configuration module



Cargo Actual: CEO & Founder

Empresa Actual: Galileo Studio

Historial Laboral: CEO & Founder at Galileo Studio (Jan 2023 - P
Ver más

Bio o Descripción: Making AI accessible for everyday use.

Nombre y Apellidos: Daniel Bicand Fernández

Tutorial de WorkOps

Tiempo estimado: 10 mins

Bienvenido/a al tutorial interactivo de **WorkOps**. Este documento está diseñado para guiarte paso a paso en el uso de la plataforma, enseñándote las funciones clave de la manera más práctica posible.

Cada sección incluye instrucciones, acciones y preguntas para el agente que se ejecutarán dentro de este mismo documento.

1. Organización: Subida y análisis de documento

Paso 1: Descargar tu perfil de LinkedIn

Para comenzar, necesitas un documento de entrada. Vamos a trabajar con tu perfil de LinkedIn:

1. Ve a tu perfil de LinkedIn.
2. Presiona `Ctrl + P` o `Cmd + P`.
3. Elige la opción "Imprimir como PDF".
4. Guarda el archivo.

¿No tienes LinkedIn? Puedes usar este el mío de ejemplo: Pablo García Bolívar

Hola, soy tu asistente para el caso "Tutorial de WorkOps". ¿En qué puedo ayudarte?

Figure 5.5 WorkOps platform: document analysis in action

Summary

These three use cases illustrate the adaptability and scalability of the document intelligence platform across domains with varying degrees of complexity and automation requirements. From highly specialized legal workflows to multilingual government correspondence and configurable SaaS tooling, the system has proven capable of delivering measurable improvements in accuracy, efficiency, and operational transparency.

In each case, the platform’s modular architecture and schema-based design allowed for rapid deployment, while the ensemble and fine-tuning capabilities enabled domain-specific optimization when necessary. These integrations confirm the practical viability of the solution and support its further adoption in enterprise-grade document processing scenarios.

Chapter 6. CONCLUSIONS

One of the clearest conclusions to emerge from this project is that building an AI-powered document processing system that works technically is relatively easy. Thanks to modern tools like GPT-4, LangChain, and cloud-based infrastructure, creating a minimum viable product (MVP) that demonstrates key functionality can be achieved in a matter of days or weeks. However, transforming that MVP into a **robust, secure, and enterprise-ready solution** is an entirely different challenge—one that requires a shift in mindset, not just in tooling.

This challenge reflects a classic **Pareto principle** in software engineering and AI: 80% of the visible functionality can be built with 20% of the total effort, but the remaining 20%—which includes reliability, maintainability, and compliance—demands 5x the time, care, and resources. This was the case in our project, where building the prototype was fast, but turning it into something deployable in real-world enterprise environments revealed much deeper layers of complexity.

Four main lessons stand out:

1. **Business Adaptation Is Underrated**
   One of the most difficult and often underestimated tasks was translating real business needs into technical abstractions. Understanding what users want is not enough; it's about defining exactly how those needs should be represented in schemas, validation logic, and model prompts. This requires close collaboration with non-technical stakeholders and a continuous feedback loop. No off-the-shelf LLM will do this on its own.
2. **Robustness Requires Infrastructure, Not Just Models**
   A working demo can give the illusion that the problem is solved. But robust systems require layers of support infrastructure: validation checks, logging mechanisms, dynamic fallback paths, error detection, and performance monitoring. More

importantly, they require explainability—being able to say why a field was extracted a certain way, or why a classification failed. These are not optional features in production—they are essential for trust and accountability.

3. **Data Security and Regulatory Compliance Are Non-Negotiable**
   When documents contain sensitive information—personal IDs, legal statements, financial data—the system cannot treat them as generic inputs. GDPR compliance, role-based access, audit trails, and infrastructure isolation all become essential components of the design. These constraints often dictate technical choices—such as keeping processing within a specific cloud provider or restricting model endpoints—even if they come at the cost of convenience or raw performance.

4. **The Human-in-the-Loop is Not a Temporary Compromise—It's a Design Pattern**
   It's tempting to think of human validation as a temporary phase before "full automation." But in reality, many domains—especially legal, finance, and compliance—require human oversight by design. This project highlights that building for human-in-the-loop workflows from the start—through UI hooks, flagging mechanisms, and review pipelines—not only improves trust but also accelerates adoption. Rather than aiming to eliminate human intervention, enterprise AI should aim to collaborate with humans by giving them visibility, control, and contextual understanding. In this sense, human-in-the-loop is not a limitation; it is a strategic design principle.

In short, this project reinforced a crucial insight: building AI tools is not just about machine learning or APIs—it's about systems thinking. Every real-world deployment requires a thoughtful balance between technical capabilities, organizational needs, and ethical responsibilities.

Chapter 7. BIBLIOGRAPHY

Artsyl. (n.d.). Retrieved from https://www.artsyltech.com/industries/document-processing-in-banking-industry

AWS. (n.d.). *AWS Products*. Retrieved from https://aws.amazon.com/es/

Edmondson, J. (2018). Retrieved from https://www.linkedin.com/pulse/ten-questions-law-firms-should-asking-themselves-client-joe-edmondson/

Emerj. (2019). Retrieved from https://emerj.com/information-extraction-in-banking/

Extend. (n.d.). *Extend AI*. Retrieved from https://www.extend.ai/

FastAPI. (n.d.). *FastAPI*. Retrieved from https://fastapi.tiangolo.com/

Invofox. (n.d.). *Invofox*. Retrieved from https://www.invofox.com/es

LangChain. (n.d.). *LangChain*. Retrieved from https://www.langchain.com/

M-Files. (2019). *M-Files.* Retrieved from https://www.m-files.com/wp-content/uploads/2023/07/ebook-2019-intelligent-information-management-benchmark-en.pdf.pdf

Mistral. (n.d.). *Mistral*. Retrieved from https://mistral.ai/news/mistral-ocr

Nanonets. (n.d.). *Nanonets*. Retrieved from https://nanonets.com/

Newswire, P. (2021). Retrieved from https://www.prnewswire.com/news-releases/intelligent-document-processing-idp-in-pc-insurance-301372490.html

OpenAI. (n.d.). *OpenAI API Docs - GPT Models*. Retrieved 06 27, 2023, from https://platform.openai.com/docs/guides/gpt

Pydantic. (n.d.). *Pydantic*. Retrieved from https://docs.pydantic.dev/latest/

Reducto. (n.d.). *Reducto AI*. Retrieved from https://reducto.ai/

TableFlow. (n.d.). *TableFlow*. Retrieved from https://tableflow.com/

United Nations. (n.d.). *United Nations Sustainable Development Goals*. Retrieved 06 27, 2023, from https://sdgs.un.org/goals

ANNEX I: SDGS ALIGNMENT

We live in a globalized world where everyone is aware of what is happening in every corner of the globe. It would be inhumane to turn a blind eye to the existing problems, especially in less developed countries. That is why, in September 2015, world leaders adopted a set of global goals to eradicate poverty, protect the planet, and ensure prosperity (United Nations, n.d.). Each goal has specific targets that must be achieved in the next 15 years.

These Sustainable Development Goals (SDGs) serve as a moral framework for every project or initiative undertaken, with them in mind, especially in an area with sufficient potential to generate a positive impact, such as computer science.

This Master's Thesis primarily aligns with Goals 8 and 9.

Goal 8

Goal 8 focuses on "promoting sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all" (United Nations, n.d.). This Master's Thesis particularly contributes to the achievement of Targets 8.2 and 8.3, which aim to achieve higher economic productivity through technological innovation and promote the growth of small and medium-sized enterprises (SMEs), respectively.si

The developed tool enables document analysis and data extraction for small and medium-sized enterprises that may not have the resources to invest in extensive software development. By not requiring programming knowledge to be used and not relying on large amounts of manually annotated documents for training, the tool can be utilized by users with varying levels of computer literacy. Thus, it provides access to a wide range of users, helping them achieve greater business productivity and fostering growth.

By enabling easier data analysis and extraction for SMEs, this tool aligns with the objectives of promoting economic growth, technological innovation, and the development of small businesses. It contributes to creating a more inclusive and sustainable economic environment, as envisioned by Goal 8 of the Sustainable Development Goals.

Goal 9

Goal 9 aims to "build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation" (United Nations, n.d.). Specifically, the developed tool aligns with Targets 9.5 and 9.b. Target 9.5 seeks to enhance the technological capabilities of industrial sectors, while Target 9.b focuses on supporting the development of technologies, research, and innovation in developing countries. The developed tool contributes to the achievement of these targets by putting an automatic data extraction tool, built with Artificial Intelligence models, into the hands of users of all types.

By providing users with a user-friendly tool for data extraction, the developed tool enhances technological capabilities and promotes the adoption of innovative solutions in various industries. It enables users, including those in developing countries, to leverage the power of Artificial Intelligence without requiring extensive technical expertise. This aligns with the goals of fostering innovation and technological advancements as outlined in Goal 9 of the Sustainable Development Goals.

8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

Figure I.0.1 SDGs - 8 and 9